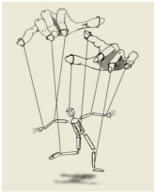


System Automation mit Puppet und Foreman

Toni Schmidbauer

5. April 2014

whoami

- SysAdmin@s-itsolutions
- toni@stderr.at
- http://github.com/tosmi
- stderr@jabber.org

Agenda

- Kurze Umfrage
- Was ist Puppet?
- Was ist Foreman?
- Puppet@s-iTSolutions
- Was haben wir geplant?

Umfrage

Was ist Puppet?

- Declarative programming: telling the machine what you would like to happen, and let the computer figure out how to do it.
- Imperative programming: telling the achine how to do something

```
class linuxwochen2014 (
  $ensure = present
) {
  user { 'toni':
    ensure => $ensure,
    uid    => 4711,
    gid    => 100,
  }

  package { 'emacs-nox':
    ensure => installed
  } ->
  package { 'vim-enhanced':
    ensure => absent,
  }
}
```

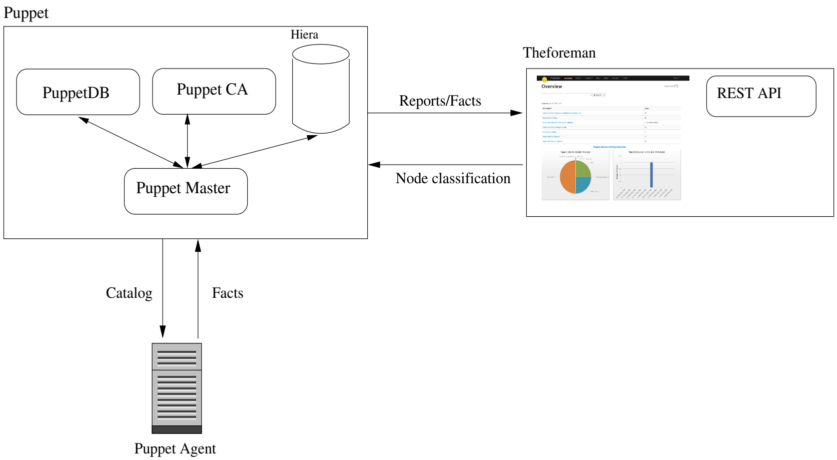
Zuordnung von Klassen

- über manifests/site.pp

```
node node /^(foo|bar)\.linuxwochen\.at$/ {
  class { 'linuxwochen2014':
    ensure => absent
  }
}
```

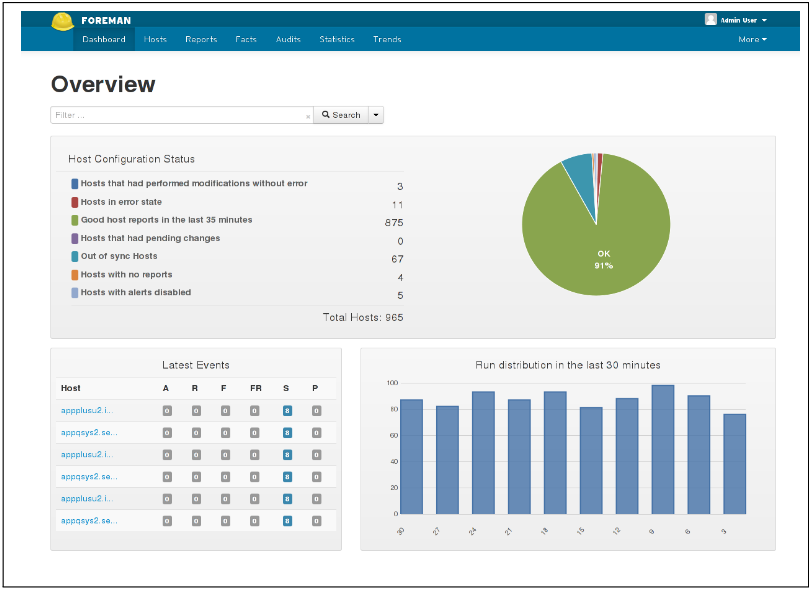
- über einen External Node Classifier (Foreman)
- über Hiera (hiera_include('classes',[""]))

Puppet run



Puppet
Hiera
PuppetDB
Puppet CA
Puppet Master
Reports/Facts
Node classification
Theforeman
Overview
REST API
Catalog
Facts
Puppet Agent

Was ist Foreman?



Overview

Host Configuration Status

🟦 Hosts that had performed modifications without error	3
🟥 Hosts in error state	11
🟩 Good host reports in the last 35 minutes	875
🟪 Hosts that had pending changes	0
🟦 Out of sync Hosts	67
🟧 Hosts with no reports	4
🟦 Hosts with alerts disabled	5

Total Hosts: 965

Latest Events

Host	A	R	F	FR	S	P
appplusu2.i...	0	0	0	0	8	0
appqsys2.se...	0	0	0	0	8	0
appplusu2.i...	0	0	0	0	8	0
appqsys2.se...	0	0	0	0	8	0
appplusu2.i...	0	0	0	0	8	0
appqsys2.se...	0	0	0	0	8	0

Run distribution in the last 30 minutes

Und jetzt?

Good

- Wie soll eine Entwicklungsumgebung aussehen?
- Wie testen wir den Puppet Code?
- Wie verwalten wir unseren Puppet Code?
- Wie soll unsere Puppet Umgebung aussehen?
- Wie erfolgt das Deployment des Codes?
- Wie verwalten wir Module von PuppetForge?

Wie soll eine Entwicklungsumgebung aussehen?

Vagrant

- `http://vagrantup.com`
- Ermöglicht virtuelle Entwicklungsumgebungen
- Vagrant Box ist ein vorkonfiguriertes Image
- Default VirtualBox andere Provider via Plugins (VMWare, KVM)

Demo

Wie testen wir den Puppet Code?

rspec-puppet

- Ruby RSpec Tests für Puppet
- Jedes Module muss RSpec Tests mitbringen

```ruby
require 'spec_helper'
describe 'linuxwochen2014' do
  let :facts { { :osfamily => 'RedHat' } }

  context 'ensure is set to absent' do
    let :params { { :ensure => 'absent'} }

    it do
      should contain_user('toni').with({
                                   'ensure' => 'absent',
                                   'uid'    => '4711',
                                   'gid'    => '100',
                                 })
    end

    it { should contain_package('emacs-nox').with_ensure('installed') }
    it { should contain_package('vim-enhanced').with_ensure('absent') }
    it { should contain_package('emacs-nox).that_comes_before('Package[vim-enhanced]') }
  end
end
```

Demo

Wie verwalten wir unseren Puppet Code?

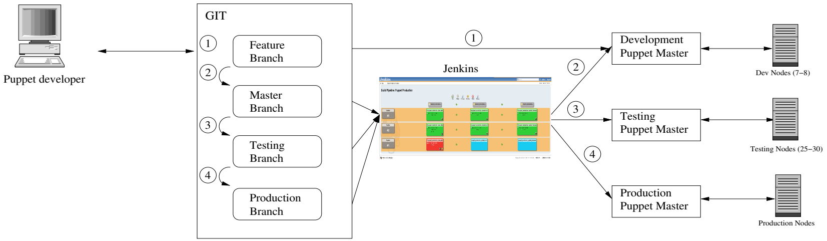
GIT

- Ein zentrales GIT Repository
- Berechtigungssystem mit Gitolite
- Feature Branches für neue Module
- 3 Hauptbranches
  - Master: Staging via GIT pull auf 4 Dev Server
  - Testing: ca. 25 “Produktions” Server (git pull)
  - Production: der Rest, Staging via tags

Wie soll unsere Puppet Umgebung aussehen?

Wie erfolgt das Deployment des Codes?

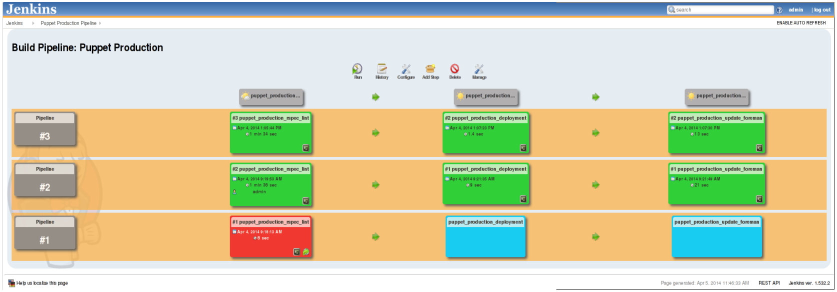
Puppet Umgebung



1. Features Branches get automatically created on Puppet Master (Dynamic Environments)
2. Master Branch gets deployed on commit via Jenkins
3. Testing Branch gets deployed via GIT tag (test_*)
   a normal commit to the Testing branch only runs tests
4. Production Branch gets deployed via GIT tag (prod_*)
   a normal commit to the Prodcution branch only runs tests

It's all the same for Hiera yaml files, except dynamic environments!

Deployment



Jenkins

search | admin | log out

Jenkins › Puppet Production Pipeline ›

ENABLE AUTO REFRESH

Build Pipeline: Puppet Production

puppet_production... → puppet_production... → puppet_production...

Pipeline #3
- #3 puppet_production_rspec_lint — Apr 4, 2014 1:05:44 PM — 1 min 34 sec
- #2 puppet_production_deployment — Apr 4, 2014 1:07:23 PM — 1.4 sec
- #2 puppet_production_update_foreman — Apr 4, 2014 1:07:30 PM — 13 sec

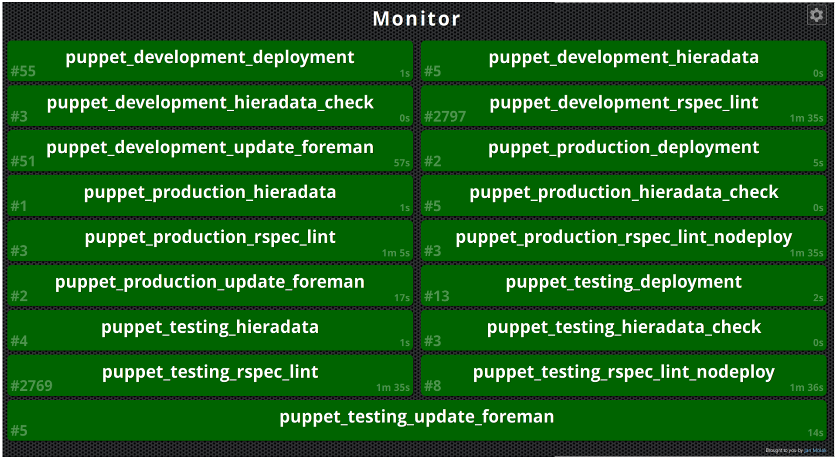
Pipeline #2
- #2 puppet_production_rspec_lint — Apr 4, 2014 9:19:53 AM — 1 min 36 sec — admin
- #1 puppet_production_deployment — Apr 4, 2014 9:21:35 AM — 9 sec
- #1 puppet_production_update_foreman — Apr 4, 2014 9:21:49 AM — 21 sec

Pipeline #1
- #1 puppet_production_rspec_lint — Apr 4, 2014 9:18:13 AM — 6 sec
- puppet_production_deployment
- puppet_production_update_foreman

Help us localize this page

Page generated: Apr 5, 2014 11:46:33 AM REST API Jenkins ver. 1.532.2

Monitoring



Monitor

puppet_development_deployment #55 1s

puppet_development_hieradata #5 0s

puppet_development_hieradata_check #3 0s

puppet_development_rspec_lint #2797 1m 35s

puppet_development_update_foreman #51 57s

puppet_production_deployment #2 5s

puppet_production_hieradata #1 1s

puppet_production_hieradata_check #5 0s

puppet_production_rspec_lint #3 1m 5s

puppet_production_rspec_lint_nodeploy #3 1m 35s

puppet_production_update_foreman #2 17s

puppet_testing_deployment #13 2s

puppet_testing_hieradata #4 1s

puppet_testing_hieradata_check #3 0s

puppet_testing_rspec_lint #2769 1m 35s

puppet_testing_rspec_lint_nodeploy #8 1m 36s

puppet_testing_update_foreman #5 14s

Brought to you by Jan Molak

Wie verwalten wir Module von PuppetForge?

Puppetforge Module

- Eigenes GIT Repository (puppetforge.git)
- Download der Module in der Enwicklungsumgebung
- Staging GIT pull (bäh!)
- Dies ändert sich allerdings (dazu später)

- Wie soll eine Entwicklungsumgebung aussehen? *DONE*
- Wie testen wir den Puppet Code? *DONE*
- Wie verwalten wir unseren Puppet Code? *DONE*
- Wie soll unsere Puppet Umgebung aussehen? *DONE*
- Wie erfolgt das Deployment des Codes? *DONE*
- Wie verwalten wir Module von PuppetForge? *DONE*

Probleme, Probleme, Probleme...

- Ein GIT Repo funktioniert nicht bei Änderungen von Upstream Modulen
- Andere Abteilungen sollen ihre Module unabhänging testen
- Unittests sagen noch nichts aus wie sich der Code am Live-System verhält
- Wir sollten eigentlich das Zusammenspiel aller Module testen (Forge und eigene)

Was haben wir geplant?

- r10k für Deployment (`https://github.com/adrienthebo/r10k`)
- Ein Repository pro Module
- Nur interne Module bleiben im Hauptrepo
- Acceptance Tests mit Beaker